ENCAPSULATING REACTING BEHAVIOUR IN GOAL-BASED PLANS FOR PROGRAMMING BDI AGENTS

Rafael H. **Bordini**
School of Technology, PUCRS
Porto Alegre, RS, Brazil
rafael.bordini@pucrs.br

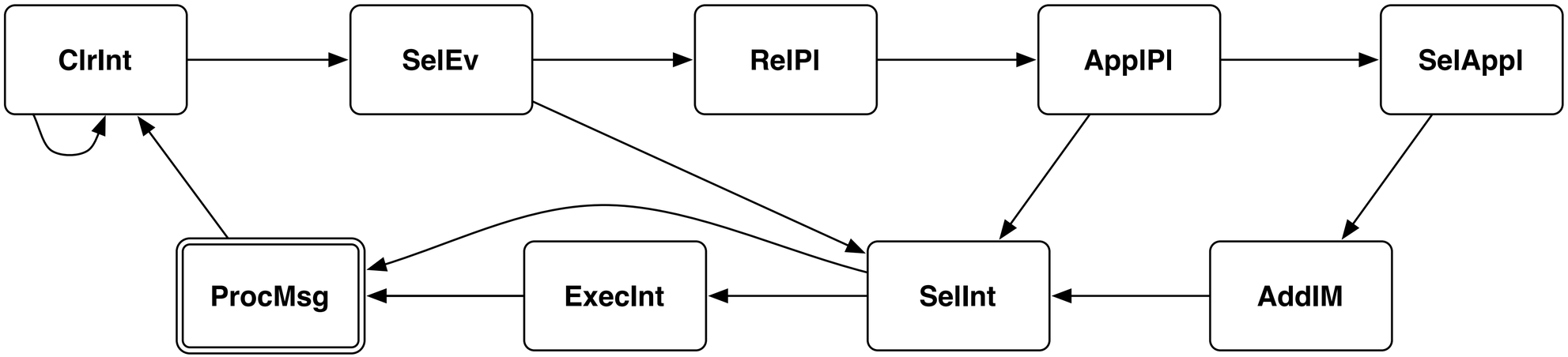
Rem **Collier**
University College of Dublin
Dublin, Ireland
rem.collier@ucd.ie

Jomi F. **Hübner**
DAS, Fed. Univ. of Santa Catarina
Florianópolis, SC, Brasil
jomi.hubner@ufsc.br

Alessandro **Ricci**
DISI, University of Bologna
Cesena, Italy
a.ricci@unibo.it

LAMAS@AAMAS2020

Reasoning Cycle



Dropping Achieved Intentions

$$\frac{\begin{array}{c} i \in C_I \qquad i = [p_1, \ldots, p_n] \\ ag_{bs} \models \mathrm{GCond}(p_j) \text{ for some } j, 1 \leq j \leq n \end{array}}{\langle ag, C, M, T, \mathrm{ClrInt} \rangle \longrightarrow \langle ag, C', M, T, \mathrm{ClrInt} \rangle} \quad (\textsc{ClrInt}_1)$$

where: j is the least number in [1..n] s.t.

$$ag_{bs} \models \mathrm{GCond}(p_j)$$

$$C'_I = C_I \setminus \{i\} \cup \{[p_1, \ldots, p_{j-1}]\}$$

Handling External Events

$$\frac{T_\varepsilon = \langle te, \top \rangle \qquad T_\rho = \{\langle p_1, \theta_1, i_1 \rangle, \ldots, \langle p_n, \theta_n, i_n \rangle\}}{\langle ag, C, M, T, \text{AddIM} \rangle \longrightarrow \langle ag, C', M, T, \text{SelInt} \rangle} \quad (\textsc{ExtEv})$$

$$\textit{where:} \qquad C'_I \quad = \quad C_I \setminus \bigcup_{j=1}^{n} \{i_j\} \cup \bigcup_{j=1}^{n} \{i_j[p_j\theta_j]\}$$

- The *RelPlans* and *AppPlans* functions also need to be adapted

Abstract Syntax

$$
\begin{array}{lll}
ag & ::= & bs\ gs\ ps \\
bs & ::= & b_1\ \ldots\ b_n \qquad (n \geq 0) \\
gs & ::= & g_1\ \ldots\ g_n \qquad (n \geq 0) \\
ps & ::= & p_1\ \ldots\ p_n \qquad (n \geq 1) \\
p & ::= & g\ f_c\ f_g\ h\ pr_1\ \ldots\ pr_n \qquad (n \geq 0) \\
r & ::= & (+b \mid -b)\ f\ h \\
pr & ::= & p \mid r \\
t & ::= & g \mid +b \mid -b \\
h & ::= & d_1\ \ldots\ d_n \qquad (n \geq 0) \\
d & ::= & a \mid g
\end{array}
$$

Agent Initialisation

Algorithm 1 Agent Initialisation

Require: an initial agent program $ag = \langle bs, gs, ps \rangle$

1: $B \leftarrow bs$
2: $E \leftarrow \{\}$
3: **for all** $g \in gs$ **do**
4: $\quad E \leftarrow E \cup \{\langle g, [] \rangle\}$
5: $P \leftarrow ps$
6: $I \leftarrow \{\}$
7: **while** *true* **do**
8: $\quad S \leftarrow$ CURRENT_PERCEPTS
9: $\quad$ GENERATE_EVENTS(S)
10: $\quad action \leftarrow$ REASONING_CYCLE( )
11: $\quad$ EXECUTE(action)

Event Generation from Percepts

Algorithm 2 Event Generation from Percepts

Require: external variables B, E

```
1:  function GENERATE_EVENTS(S)
2:      ▷ S is a set of percepts from the agent's sensors
3:      for all s ∈ S do
4:          if s ∉ B then
5:              B ← B ∪ {s}
6:              E ← E ∪ {⟨+s, []⟩}
7:      for all b ∈ B do
8:          if b ∉ S then
9:              B ← B \ {b}
10:             E ← E ∪ {⟨−b, []⟩}
```

Reasoning Cycle: Achieved Goals

Algorithm 3 Reasoning Cycle

Require: external variables B, E, P, I

1: **function** REASONING_CYCLE
2: ▷ Drop all achieved goals
3: **for all** $i \in I$ **do**
4: **for** $j = \text{LENGTH}(i)$ down to 1 **do** ▷ From bottom to top
5: $i'' \leftarrow [p_j[h_j]] \in i$
6: let $p_j = \langle t, f_c, f_g, h, pr_1, \ldots, pr_n \rangle$
7: **if** $B \models f_g$ **then**
8: $I \leftarrow I \setminus \{i\} \cup \{\text{REMOVE}(i'', i)\}$

RC: Handling an Event 1

9: ▷ Handle an event
10: $se \leftarrow \textsc{select\_ev}(E)$
11: let $se = \langle t, i \rangle$
12: **if** $i = \{\} \wedge t : (+b \mid -b)$ **then** ▷ external belief event
13: **for all** $i \in I$ **do** ▷ may trigger a r.r. in each intention
14: $rps = \textsc{get\_relevant}(t, i, \{\})$ ▷ search entire i
15: $aps = \textsc{get\_applicable}(rps)$
16: **if** $aps \neq \{\}$ **then**
17: $r = \textsc{select\_opt}(aps)$
18: let $r = \langle t\, f\, h \rangle$
19: let $i = [p_1[h_1], i']$
20: $i'' \leftarrow [p_1[h, h_1], i']$
21: $I \leftarrow I \setminus \{i\} \cup \{i''\}$

RC: Handling an Event 2

22: **else if** $i = \{\} \wedge t : g$ **then** ▷ external goal event
23: $rps \leftarrow$ GET_RELEVANT$(t, [], \{\})$ ▷ search only in P
24: $aps \leftarrow$ GET_APPLICABLE(rps)
25: **if** $aps \neq \{\}$ **then**
26: $p \leftarrow$ SELECT_OPT(aps)
27: let $p = \langle t, f_c, f_g, h, pr_1, \ldots, pr_n \rangle$
28: **if** $B \not\models f_g$ **then**
29: $I \leftarrow I \cup \{[p[h]]\}$
30: **else if** $i \neq \{\} \wedge t : g$ **then** ▷ internal (goal) event
31: $rps \leftarrow$ GET_RELEVANT$(t, i, \{\})$ ▷ search in i
32: ▷ and top level plans in P
33: $aps \leftarrow$ GET_APPLICABLE(rps)
34: **if** $aps \neq \{\}$ **then**
35: $p \leftarrow$ SELECT_OPT(aps)
36: let $p = \langle t, f_c, f_g, h, pr_1, \ldots, pr_n \rangle$
37: **if** $B \not\models f_g$ **then**
38: $I \leftarrow I \setminus \{i\} \cup \{$PUSH$(p[h], i)\}$

RC: Executing an Intention

39: ▷ Execute a step of an intention
40: $i \leftarrow$ SELECT_INT(I)
41: let $i = [p_1[d_1, h_1], i']$
42: **if** $d_1 : g$ **then**
43: $E \leftarrow E \cup \{\langle d_1, i \rangle\}$
44: $action \leftarrow$ REASONING_CYCLE$(\,)$
45: **return** $action$
46: **else if** $d_1 : a$ **then**
47: $i'' \leftarrow [p_1[h_1], i']$
48: $I \leftarrow I \setminus \{i\} \cup \{i''\}$
49: **return** d_1
50: ▷ See note in the text about empty stacks

Retrieving Relevant Plans

Algorithm 4 Retrieving Relevant Plans

Require: external variable P

```
1:  function GET_RELEVANT(t,i,rps)
2:      if i ≠ {} then
3:          let p[h] = HEAD(i)
4:          for all pr ∈ p do
5:              if RELEVANT(pr,t) then
6:                  rps ← rps ∪ {pr}
            GET_RELEVANT(t, TAIL(i), rps)
7:      else
8:          for all p ∈ P do      ▷ check for relevant top-level plans
9:              if RELEVANT(p,t) then
10:                 rps ← rps ∪ {p}
11:     return rps
```

ENCAPSULATING REACTING BEHAVIOUR IN GOAL-BASED PLANS FOR PROGRAMMING BDI AGENTS

Rafael H. **Bordini**
School of Technology, PUCRS
Porto Alegre, RS, Brazil
rafael.bordini@pucrs.br

Rem **Collier**
University College of Dublin
Dublin, Ireland
rem.collier@ucd.ie

Jomi F. **Hübner**
DAS, Fed. Univ. of Santa Catarina
Florianópolis, SC, Brasil
jomi.hubner@ufsc.br

Alessandro **Ricci**
DISI, University of Bologna
Cesena, Italy
a.ricci@unibo.it

LAMAS@AAMAS2020